Phase I

On the basis of an example project and existing sketches, you should make yourself familiar with Sketchlink. This phase focuses on detecting and viewing linked sketches, as well as navigating source code with them.

Task 1

1. Start the IntelliJ Community IDE and open the project “SDRaytracer”.
2. Open the file “SDraytracer.java”.

Task 2

An icon in the Javadoc comment indicates that a linked sketch might be available.

1. Hover the mouse over the icon inside the Javadoc comment of the main class to highlight its corresponding artifact.
2. Click on the icon to open a list of available sketches.
3. Open the preview of the “Overview” sketch by hovering over its entry.
4. Hover your mouse cursor over different marked sketch areas to inspect their annotations and authors.

Task 3

1. Repeat Task 2 for the “renderimage()” method, selecting the “Overview” sketch again.
2. Click on the pin symbol in the upper right corner to switch from pop-up to tool window mode.
3. Left click on a sketch area to jump to a linked source code artifact.

Task 4

While it is only possible to view sketches inside the IDE, you can use the Sketchlink WebView to view and edit them. There are two possible ways to open a WebView with a sketch. You can click on a sketchlink icon and then left click an entry on the list. Alternatively, you can directly right click inside a sketch.

1. Make sure the sketch from the previous task is still visible.
2. Open a WebView.
3. Choose to open inside a new local web browser tab.
4. Arrange the web browser and the IntelliJ window side by side.
5. Inside the WebView, select the "lightning()" marker.
6. Use the "Follow Link" button.

Task 5

You can view a sketch inside an existing WebView client, even if isn't running on the same machine.

1. Repeat Steps 1 and 2 from Task 4.
2. Select the WebView running on the supplied iPad.
3. Use the tablet to browse through the project again.

Task 6

Explore the program using the "Overview" sketch and the Javadoc comments. Concentrate on "SDRaytrace.java" and stop when you fell confident that you understand the general structure.

Question 1: What are these source code artifacts linked to?		
	method "raytrace()"	class "SDRaytrace()"
This artifact is linked to at least three areas in this sketch.		
This artifact is linked to exactly two areas in the sketch.		
This artifact is linked to exactly one area in the sketch.		
This artifact is only linked to the sketch as a whole.		
This artifact is linked to this sketch as a whole AND atleast one of the marked areas.		
This artifact is not linked to this sketch.		
I'm unsure.		

Phase II

This phase focuses on documenting existing code with a newly created sketch.

Task 1

1. Open the project "ARC4-Example"
2. Open the file "ARC4.java".

Task 2

1. Look through the code of the method "byte[] process(byte[] data,byte password)".
2. Write a short Javadoc comment for this method, explaining **what it is supposed to do**.
3. Sketch on a piece of paper or the whiteboard to understand **how it works**.

Question 2: How helpful was sketching to you during this task?					
not helpful		→			very helpful

Question 3: How helpful do you think your sketch will be for you in the future?					
not helpful		→			very helpful

Question 4: How helpful do you think your sketch would be for others?					
not helpful		→			very helpful

Task 3

1. Use the provided tablet and capture your sketch with the WebView
2. Upload your sketch to the server

Task 4

Link your sketch to a method. The caret position decides, which code artifact is linked.

1. Inside the IDE, create a link anchor for the “process” method using the context menu.
2. Select “Show in WebView”.
3. In the WebView, open your sketch.
4. Link the whole sketch to the selected method.

Task 5

Link only a region of your sketch to a section of a method.

1. Switch back to the IDE.
2. Create a link anchor *inside* the “process” method at an important code artifact.
3. Select “Show in WebView” and open your sketch.
4. Mark a specific area of your sketch and link to it.

Question 5: How helpful do you think your sketch will be for others, when it is available next to the source code?					
not helpful		→			very helpful

Sketchlink – Questionnaire

Age:	
Gender:	

Participant ID

General Questions:

1. How much experience do you have …	no experience →					expert
… in object-oriented programming.						
… with Java.						
… with IntelliJ IDEA.						
… with ray tracing principles.						
… with the aRC4 (WEP) encryption algorithm.						
… with Sketchlink before this experiment.						

2. How would you rate the importance of sketches and diagrams in your professional work?	unimportant →					important
Please elaborate.						

Usability:

3. Please rate the tool according to the following statements:	strongly disagree →					strongly agree
I would use this system frequently.						
I find the system unnecessarily complex.						
I find the system easy to use.						
I need the support of a technical person to use this system.						
The ability to preview sketches inside the IDE is important.						
I find the system inconsistent.						
I imagine that most people would learn to use this system quickly.						
Using this system is cumbersome.						
I feel confident using the system.						
I needed to learn a lot of things before I could get going with this system.						

4. I have <u>not</u> been restricted in my individual workflow when using the tool.	strongly disagree → strongly agree
During which Tasks did you feel restricted?	

5. During Task 1, I was able to understand the overall structure of the program and its marked components more quickly due to the availability of sketches.	strongly disagree → strongly agree
Please give an example, if possible, where an available sketch helped your comprehension of source code beyond Javadoc comments.	

6. I was always able to detect which parts of the sketch were linked to a selected code artifact.	strongly disagree → strongly agree
Please describe a situation in which a more visible mark would have been desirable.	

7. I was always able to navigate to an expected code artifact using a sketch.	strongly disagree → strongly agree
If not, please contrast the expected and observed behavior.	

8. I found the icons to be discreetly integrated into the source code editor.	strongly disagree → strongly agree					
Please suggest further improvements.						

9. If a source code artifact had linked sketches available, I was always able to locate the icon providing these sketches.	strongly disagree → strongly agree					
Please describe a moment when an icon was at an unexpected position.						

10. Inside a method: Linking to a single statement, loop, or line is completely sufficient.	strongly disagree → strongly agree					
Please elaborate.						

11. Were there any features missing that would have aided in solving the tasks? Which additional features would enhance this tool?

Further Remarks:

12. During **Task 1**, I had no difficulties understanding the provided sketches.	strongly disagree → strongly agree					
Please describe those situations where difficulties arose.						

13. During **Task 1**, I had no difficulties understanding the provided source code.	strongly disagree → strongly agree					
Please give examples of methods you found to be obscure.						

14. During **Task 2**, I had no difficulties understanding the provided source code.	strongly disagree → strongly agree					
Please explain what difficulties you had.						

15. Other remarks